

DAM  
Desarrollo de Aplicaciones Multiplataforma  
2º Curso

AD  
Acceso a Datos

UD 7  
Bases de Datos noSQL

IES BALMIS  
Dpto Informática  
Curso 2019-2020  
Versión 1 (09/2019)

## 4. EJERCICIOS con MongoDB desde Java (RedSocial)

Utiliza la consola de MongoDB para crear la base de datos **redsocial** como se ha visto en los ejemplos y ejecuta las instrucciones del archivo **redsocial.js** desde Consola para crear la colección **usuario**.

### CONSULTAS

#### RedSocial01Consultas

Hacer un proyecto que contenga un menú con las siguientes opciones:

- **4101. Obtener las mujeres con estado activo.**  
Mostrar los campos Código(\_id), Nombre, Apellidos, Sexo y Activo  
Podéis crear el filtro de dos formas:
  - Mediante un objeto json (Document.parse) que puede probarse en MongoDB console
  - Mediante Filters y sus métodos and() y eq().
- **4102. Obtener los hombres que tienen más de un mensaje.**  
Mostrar los campos Código(\_id), Nombre, Apellidos, Sexo y Activo  
En este caso podéis crear el filtro de dos formas:
  - Mediante un objeto json (Document.parse)
  - Mediante los métodos and(), gt() y eq().
- **4103. Obtener usuarios que pertenecen a los grupos de futbol y cine (a uno de los dos)**  
Mostrar los campos Código(\_id), Nombre, Apellidos
- **4104. Obtener usuarios que pertenecen a los grupos de futbol y cine (a los dos)**  
Mostrar los campos Código(\_id), Nombre, Apellidos

```
-----
1. Mujeres (activo=true)
2. Hombres (total_mensajes>1)
3. Usuarios de fútbol o cine
4. Usuarios de fútbol y cine
-----
0. Salir
-----
```

Escribe una de las opciones: 1

Mostrar Mujeres en estado activo

```
=====
+-----+-----+-----+-----+-----+
|Codigo|Nombre|Apellidos|Sexo|Activo|
+-----+-----+-----+-----+-----+
|3|Laura|Górriz Pardo|M|SI|
|5|María|Soriano Martín|M|SI|
|8|Marta|Fernández García|M|SI|
+-----+-----+-----+-----+-----+
```

```
-----
1. Mujeres (activo=true)
2. Hombres (total_mensajes>1)
3. Usuarios de fútbol o cine
4. Usuarios de fútbol y cine
-----
0. Salir
-----
```

Escribe una de las opciones: 2

Mostrar Mujeres en estado activo

```
=====
+-----+-----+-----+-----+-----+
|Codigo|Nombre|Apellidos|Sexo|Mensajes|
+-----+-----+-----+-----+-----+
|2|Arturo|Cuenca Bertolín|V|2|
+-----+-----+-----+-----+-----+
```

- ```

-----
1. Mujeres (activo=true)
2. Hombres (total_mensajes>1)
3. Usuarios de fútbol o cine
4. Usuarios de fútbol y cine
-----

```

```
0. Salir
-----
```

```
Escribe una de las opciones: 3
-----
```

```
Mostrar Usuarios de grupos fútbol o cine
=====
```

| Codigo | Nombre    | Apellidos        |
|--------|-----------|------------------|
| 1      | Ramón     | García Trueca    |
| 2      | Arturo    | Cuenca Bertolín  |
| 3      | Laura     | Górriz Pardo     |
| 4      | Alejandro | León Córdoba     |
| 5      | María     | Soriano Martín   |
| 8      | Marta     | Fernández García |

- ```

-----
1. Mujeres (activo=true)
2. Hombres (total_mensajes>1)
3. Usuarios de fútbol o cine
4. Usuarios de fútbol y cine
-----

```

```
0. Salir
-----
```

```
Escribe una de las opciones: 4
-----
```

```
Mostrar Usuarios de grupos fútbol y cine
=====
```

Codigo	Nombre	Apellidos
1	Ramón	García Trueca
5	María	Soriano Martín

## AGRUPACIONES

### RedSocial02Agrupaciones

Hacer un proyecto que contenga un menú con las siguientes opciones:

- **4201. Mostrar para cada usuario los datos de sus amigos (mostrar sólo de cada usuario y sus amigos el id y el nombre)**
  - Nota1: el campo amigos es un ArrayList de Long
  - Nota2: antes de recorrerlo habrá que comprobar primero que no es null para evitar la excepcion NullPointerException
- **4202. Mostrar el usuario más joven.**  
 Consulta MongoDB:  

```
db.usuario.find({},
                    {nombre:1,apellidos:1,nacimiento:1}
                ).sort({nacimiento:-1}).limit(1).pretty()
```
- **4203. Mostrar los países, sus ciudades y el número de usuarios que tiene, ordenado por país y luego por ciudad.**
  - Nota1: utilizaremos aggregate en vez de find.
  - Nota2: el filtro es un ArrayList<Bson> en vez de un Bson
- **4204. Mostrar los nombres de usuarios de cada sexo**
  - Nota1: Debes realizar un recorrido sobre la consulta de sexos (utilizando aggregate) y luego un recorrido por sus usuarios (utilizando find), es decir dos bucles

```
-----
1. Amigos de usuarios
2. Usuario más joven
3. Países y ciudades
4. Nombres de cada sexo
```

```
-----
0. Salir
```

```
-----
Escribe una de las opciones: 1
```

```
-----
Mostrar Amigos de Usuarios
```

```
=====
```

```
-----
1 Ramon
=> 2 Arturo
```

```
2 Arturo
=> 1 Ramón
```

```
3 Laura
```

```
4 Alejandro
=> 1 Ramón
=> 5 María
```

```
5 Maria
=> 8 Marta
=> 4 Alejandro
```

```
6 Gabriel
```

```
7 Daniela
```

```
8 Marta
=> 5 María
```

```
-----
```

- ```

-----
1. Amigos de usuarios
2. Usuario más joven
3. Países y ciudades
4. Nombres de cada sexo
-----

```

```
0. Salir
-----

```

```
Escribe una de las opciones: 2
-----

```

```
Mostrar el usuario más joven
=====

```

| Codigo | Nombre  | Apellidos     | Nacimiento |
|--------|---------|---------------|------------|
| 6      | Gabriel | Torres García | 07/04/2002 |

- ```

-----
1. Amigos de usuarios
2. Usuario más joven
3. Países y ciudades
4. Nombres de cada sexo
-----

```

```
0. Salir
-----

```

```
Escribe una de las opciones: 3
-----

```

```
Países y ciudades
=====

```

País	Ciudad	Núm. Usuarios
España	Valencia	3
España	Castellón	3
Venezuela	Valencia	1
España	Alicante	1

- ```

-----
1. Amigos de usuarios
2. Usuario más joven
3. Países y ciudades
4. Nombres de cada sexo
-----

```

```
0. Salir
-----

```

```
Escribe una de las opciones: 4
-----

```

```
Mostrar Usuarios de cada sexo
=====

```

```
SEX0: M
=====

```

| Codigo | Nombre  | Apellidos        |
|--------|---------|------------------|
| 7      | Daniela | Serna Ortiz      |
| 3      | Laura   | Górriz Pardo     |
| 5      | María   | Soriano Martín   |
| 8      | Marta   | Fernández García |

```
SEX0: V
=====

```

| Codigo | Nombre    | Apellidos       |
|--------|-----------|-----------------|
| 4      | Alejandro | León Córdoba    |
| 2      | Arturo    | Cuenca Bertolín |
| 6      | Gabriel   | Torres García   |
| 1      | Ramón     | García Trueca   |

## ACTUALIZACIONES

### RedSocial03Actualizaciones

Hacer un proyecto que contenga un menú con las siguientes opciones:

- 4301. todos los usuarios hay que quitarle el campo "fecha"  
Update en MongoDB:  

```
db.usuario.update({}, {$unset:{"fecha":1}})
```
- 4302. A todos los usuarios añadirle el campo cantidad\_amigos.
- 4303. Añadir el comentario el siguiente comentario a Daniela:  

```
{
  "titulo" : "La casa de papel",
  "texto" : "Excelente serie española",
  "fecha" : ISODate("2019-09-11T11:30:25Z"),
  "grupo" : "cine"
}
```

  
Nota1: Ten en cuenta que también habrá que incrementar el contador de comentarios (total\_comentarios)

```
-----
1. Quitar campos fecha
2. Añadir campos cantidad_amigos
3. Añadir comentario a Daniela
-----
0. Salir
-----
Escribe una de las opciones: 1
-----
Eliminar a todos los usuarios el campo fecha
=====
Eliminado
-----
1. Quitar campos fecha
2. Añadir campos cantidad_amigos
3. Añadir comentario a Daniela
-----
0. Salir
-----
Escribe una de las opciones: 2
-----
Añadir campo cantidad de amigos
=====
Cantidad de amigos actualizada
-----
1. Quitar campos fecha
2. Añadir campos cantidad_amigos
3. Añadir comentario a Daniela
-----
0. Salir
-----
Escribe una de las opciones: 3
-----
Añadir comentario a Daniela
=====
Comentario añadido
```

## ACTUALIZACIONES

### RedSocial04Funciones

Hacer un proyecto que contenga un menú con las siguientes opciones:

- 4401. Implementar una función que reciba como parámetro el nombre de un grupo y devuelva la cantidad de usuarios de dicho grupo.
  - Añadir la opción de menú que pida el grupo y llame a la función para mostrar la cantidad.
- 4402. Implementar una función que reciba el nombre de un usuario y muestre sus amigos (mostrar el nombre).
  - Añadir la opción de menú que pida el nombre y llame a la función para mostrar sus amigos.
- 4403. Implementar una función que reciba el nombre de dos personas (sabemos que el nombre no se repite), y que se encargue de añadir cada persona al campo amigos de la otra persona.
  - Se mostrará un mensaje de error:
    - Si alguno de los dos nombres de persona no existe
    - Si los dos nombres son iguales
    - Si está repetido o ya incluye como amigo al otro se mostrará el mensaje de error correspondiente.
  - Añadir una opción de menú que pida los nombres y llame a la función. Luego utilizando la función anterior, que muestre los amigos de los dos.

```
-----
1. Número de miembros de un grupo
2. Amigos de un usuario
3. Hacer amigos
-----
0. Salir
-----
Escribe una de las opciones: 1
-----
Introduzca el nombre de un grupo: cine
El número de miembros del grupo 'cine' es 5
```

```
-----
1. Número de miembros de un grupo
2. Amigos de un usuario
3. Hacer amigos
-----
0. Salir
-----
Escribe una de las opciones: 2
-----
Introduzca el nombre de un usuario: Alejandro
Los amigos de 'Alejandro' son:
=> 1 Ramón
=> 5 María
```

```
-----
1. Número de miembros de un grupo
2. Amigos de un usuario
3. Hacer amigos
-----
0. Salir
-----
Escribe una de las opciones: 3
-----
Introduzca el nombre de dos usuarios que serán amigos.
Usuario 1: Marta
Usuario 2: Daniela
Los usuarios se han actualizado como amigos.
Los amigos de 'Marta' son:
=> 5 María
=> 7 Daniela

Los amigos de 'Daniela' son:
=> 8 Marta
```

```
-----
1. Número de miembros de un grupo
2. Amigos de un usuario
3. Hacer amigos
-----
0. Salir
-----
Escribe una de las opciones: 3
-----
Introduzca el nombre de dos usuarios que serán amigos.
Usuario 1: Ramón
Usuario 2: Arturo
El usuario 'Arturo' ya es amigo de 'Ramón'
El usuario 'Ramón' ya es amigo de 'Arturo'
```